A Comprehensive Study of Decoder-Only LLMs for Text-to-Image Generation

Supplementary Material

A. Training details

We follow the U-Net [8] based latent diffusion architecture from Stable Diffusion v2 [7] with a replication training framework by MosaicML [11]. We use Diffusers as our main model codebase for the Variational Autoencoder (VAE), U-Net, and noise scheduler [12]. The configurations all follow the original Stable Diffusion [7]. For each model, we swap the text encoder and freeze all components except for the U-Net. Before the text embeddings are input to the U-Net's cross-attention blocks, we apply a linear projection from each text encoder's embedding dimension to 1024 output features for all models. The base U-Net has 865,910,724 trainable parameters, and the additional parameters from the projection for each model are shown in Table A.

For training, we use a 46 million text-image pair subset of the LAION-Aesthetics dataset [9]. We perform center cropping on all training images and LLM-based caption upsampling with VisualFactChecker (VFC) [2]. We train all models for 800,000 iterations at 256×256 resolution with a global batch size of 2,048 on 32 A100 GPUs. We use a caption drop probability of 0.1 and use the AdamW optimizer [5] with a learning rate of 10^{-4} and weight decay of 0.01. We also use additional optimizations such as FlashAttention [1], half precision GroupNorm, half precision LayerNorm, and Fully Sharded Data Parallel's (FSDP) [13] *SHARD_GRAD_OP* mode for enhanced GPU scaling [11]. Notably, we also pre-compute the VAE image latents and text embeddings for the entire training dataset to reduce repetitive compute time and costs, with each model taking around 7 days to fully train.

B. Prompt upsampling

B.1. Prompting

We use the instruction fine-tuned Gemma2-9B [10] to perform prompt upsampling at inference. We use the following prompt:

```
You are part of a team that creates images
    from texts. You work with an assistant
    that will draw anything you say. For
    example, outputting "a beautiful morning
     in the woods with the sun peaking
    through the trees" will trigger your
    partner to output an image of a forest
    morning, as you described.

You are responsible for helping people
    create detailed, amazing images. The way
     to accomplish this is to take their
    short prompts and make them extremely
    detailed and descriptive.

There are a few rules to follow:
- Try to start with the original input
    prompt in your output, and then add
    details after.
- You will only ever output a single image
    description per user request.
- Image descriptions must be between 15-80
    words. Extra words will be ignored.
```

With the following generation parameters:

```
do_sample=True
temperature=0.01
top_p=1.0
max_new_tokens=1024
```

B.1.1. Examples

In this subsection, we list a few examples of the upsampled prompts from GenAI-Bench [4] that are used in Figure 4:

A tomato vine with several tomatoes on it, all yellow except the largest which is red ⇒ "A tomato vine with several tomatoes on it, all yellow except the largest which is red. The vine is thick and green, with leaves that are a deep, rich green. The tomatoes are plump and juicy, glistening in the sunlight. The largest tomato, a vibrant red, hangs heavy on the vine, contrasting beautifully with the yellow fruit around it."

A tomato vine with several tomatoes on it, all yellow except the largest which is red ⇒ "A larger gorilla hands a smaller mechanical monkey a banana. The gorilla is silverback, with thick fur and a wise expression in its eyes. The mechanical monkey is made of polished brass, with intricate gears visible on its chest and limbs. It has glowing red eyes and a mischievous grin. The banana is ripe and yellow, held out in the gorilla's massive hand. The background is a lush jungle, with vines and ferns creating a vibrant tapestry."

C. VQAScore evaluation

C.1. VQAScore details

VQAScore is a metric for evaluating how well a generated image semantically aligns with its text prompt by using visual-question-answering (VQA). It ranges from 0 to 1, where scores closer to 1 represent close alignment with the prompt, and 0 means the generated image doesn't at all. Intuitively, it measures how well the image understands and

Text Encoders	Embedding Dimension	Extra Parameters
CLIP, T5	1024	1,048,576
Qwen2-1.5B	1536	1,572,864
Gemma2-2B	2304	2,359,296
Qwen2-7B, Gemma2-9B, gte-Gwen2, bge-Gemma2	3584	3,670,016
Mistral-7B, Llama3-8B, sfr-Mistral, Mistral-Instruct	4096	4,194,304

Table A. Additional trainable parameters from adding a linear projection layer from text encoder's embedding dimensions to 1024 output features before cross-attention.

Model	Embeddings	Avg	Attr.	Scene	Spat.	Action	Part	Count.	Comp.	Differ.	Neg.	Uni.
T5-XXL	last layer	0.741	0.737	0.809	0.741	0.782	0.723	0.677	0.717	0.675	0.599	0.757
T5-XXL	norm avg	0.747	0.748	0.813	0.745	0.780	0.720	0.687	0.736	0.675	0.617	0.760
Qwen2-7B	last layer	0.683	0.679	0.805	0.670	0.724	0.657	0.588	0.603	0.590	0.552	0.763
Qwen2-7B	norm avg	0.740	0.741	0.823	0.740	0.772	0.731	0.680	0.704	0.683	0.589	0.739
Mistral-7B	last layer	0.675	0.667	0.763	0.665	0.711	0.641	0.576	0.556	0.526	0.524	0.726
Mistral-7B	norm avg	0.769	0.774	0.837	0.780	0.802	0.733	0.699	0.716	0.706	0.630	0.789
Llama3-8B	last layer	0.675	0.673	0.767	0.656	0.704	0.667	0.627	0.615	0.568	0.542	0.768
Llama3-8B	norm avg	0.744	0.744	0.831	0.744	0.783	0.705	0.704	0.675	0.659	0.628	0.782
Gemma2-9B	last layer	0.710	0.709	0.794	0.711	0.760	0.705	0.642	0.659	0.617	0.544	0.709
Gemma2-9B	norm avg	0.753	0.757	0.814	0.743	0.790	0.735	0.691	0.703	0.679	0.651	0.770
gte-Qwen2	last layer	0.482	0.486	0.537	0.479	0.497	0.466	0.446	0.393	0.405	0.424	0.437
gte-Qwen2	norm avg	0.654	0.647	0.746	0.626	0.696	0.632	0.539	0.619	0.536	0.538	0.683
sfr-Mistral	last layer	0.710	0.706	0.804	0.707	0.740	0.691	0.661	0.670	0.615	0.608	0.766
sfr-Mistral	norm avg	0.750	0.745	0.839	0.762	0.782	0.713	0.677	0.715	0.706	0.610	0.785
bge-Gemma2	last layer	0.737	0.730	0.824	0.729	0.793	0.722	0.662	0.654	0.641	0.623	0.797
bge-Gemma2	norm avg	**0.789**	**0.787**	**0.846**	**0.782**	**0.821**	**0.786**	**0.745**	**0.776**	**0.744**	**0.712**	**0.810**

Table B. VQAScore for models using different embedding strategies: standard last-layer embeddings (last layer) and average embeddings across all normalized layers (norm avg). Highest scores are shown in **bold**. Our results show that using layer-normalized averaging significantly enhances performance and most models outperform T5.

represents the prompt, going beyond surface-level similarity. Please refer to the original VQAScore paper [4] for further insight.

C.2. Additional layer-normalized averaging results

We report additional results for our models using layer-normalized average embeddings, which aggregate representations across all layers. Table B presents a comprehensive comparison with the baseline T5 model and models utilizing last-layer embeddings.

C.3. Original CLIP-FlanT5 model

We show results for our models evaluated using the original VQAScore implementation in Tables C, D, E, F, G. As can be seen in these tables, the custom CLIP-FlanT5 model introduced in VQAScore paper [4] is not as capable as GPT-4o [3] in discriminating between different models, but still show correlated trends to the GPT-4o results.

C.4. Our GPT-4o implementation

We build upon VQAScore's support for GPT-4v by replicating the code and swapping in the GPT-4o API instead. We also limit the number of tokens returned by setting *top_logprobs* $= 20$. We found that a simple retry up to 3 times eliminated almost all errors, such as timeout and invalid answer token selections. Apart from enhanced performance, GPT-4o also includes support for prompt caching, allowing for reduced time and costs.

C.5. Random variation

When computing VQAScore with GPT-4o, we generate the 1600 images of upsampled GenAI-Bench prompts for each

Model	Size	Avg	Attr.	Scene	Spat.	Action	Part	Count.	Comp.	Differ.	Neg.	Uni.
$\text{CLIP}_{\text{ViT-H/14}}$	354M	0.761	0.762	0.790	0.765	0.762	0.749	0.755	0.758	0.729	0.710	0.771
T5-XXL	4.7B	0.795	0.795	0.816	0.803	0.803	0.780	0.799	0.800	0.793	0.739	0.804
Qwen2-7B	7B	0.772	0.772	0.798	0.777	0.782	0.761	0.760	0.759	0.748	0.712	0.785
Mistral-7B	7B	0.767	0.765	0.787	0.771	0.771	0.751	0.756	0.753	0.733	0.710	0.780
Llama3-8B	8B	0.770	0.769	0.795	0.773	0.775	0.767	0.767	0.768	0.757	0.721	0.793
Gemma2-9B	9B	0.782	0.782	0.801	0.790	0.787	0.776	0.781	0.779	0.769	0.708	0.784
gte-Qwen2	7B	0.597	0.605	0.604	0.609	0.579	0.588	0.602	0.620	0.605	0.605	0.611
sfr-Mistral	7B	0.782	0.780	0.810	0.790	0.782	0.768	0.786	0.786	0.777	0.738	0.794
$\text{Mistral-7B}_{\text{Instruct}}$	7B	0.777	0.776	0.799	0.781	0.781	0.765	0.782	0.771	0.758	0.720	0.782
bge-Gemma2	9B	0.786	0.782	0.808	0.790	0.790	0.774	0.786	0.773	0.781	0.750	0.792

Table C. Original VQAScore for models using embeddings extracted from the last layer. We use text encoders from CLIP-ViT-H/14 (354M) and T5-XXL (4.7B), along with four popular open-source pre-trained LLMs: Qwen2 (7B), Mistral-7B (7B), Llama3 (8B), and Gemma2 (9B). Additionally, we include three embedding models fine-tuned on these LLMs: gte-Qwen2 (gte-Qwen2-7B-instruct; 7B), sfr-Mistral (SFR-Embedding-2_R; 7B), and bge-Gemma2 (bge-multilingual-gemma2; 9B). We also include an instruction fine-tuned model, Mistral-7B-Instruct (7B).

Model	Layer	Avg	Attr.	Scene	Spat.	Action	Part	Count.	Comp.	Differ.	Neg.	Uni.
T5-XXL	25 (last)	0.795	0.795	0.816	0.803	0.803	0.780	0.799	0.800	0.793	0.739	0.804
Mistral-7B	33 (last)	0.782	0.780	0.810	0.790	0.782	0.768	0.786	0.786	0.777	0.738	0.794
Mistral-7B	32	0.783	0.782	0.802	0.785	0.790	0.775	0.783	0.766	0.779	0.726	0.786
Mistral-7B	15	0.783	0.785	0.811	0.787	0.788	0.774	0.795	0.783	0.779	0.724	0.783
Mistral-7B	0 (first)	0.660	0.661	0.702	0.657	0.646	0.630	0.655	0.680	0.640	0.605	0.692

Table D. Original VQAScore for models using embeddings extracted from individual layers of Mistral-7B (7B). We include the baseline T5-XXL (4.7B) model using embeddings extracted from the last layer as a reference.

model. The variation resulting from the choice of random seeds is in the order of ± 0.004, depending on the category. The variation from the same seed is ± 0.003, from non-deterministic CUDA operations and possible non-determinism of the GPT-4o queries.

D. More visual comparisons

We provide additional visual comparisons between the baseline CLIP and T5 models using last-layer embeddings with the Mistral and bge-Gemma2 models using layer-normalized average embeddings. Figure A shows examples from common text-to-image prompts, and Figure B shows additional prompts from GenAI-Bench.

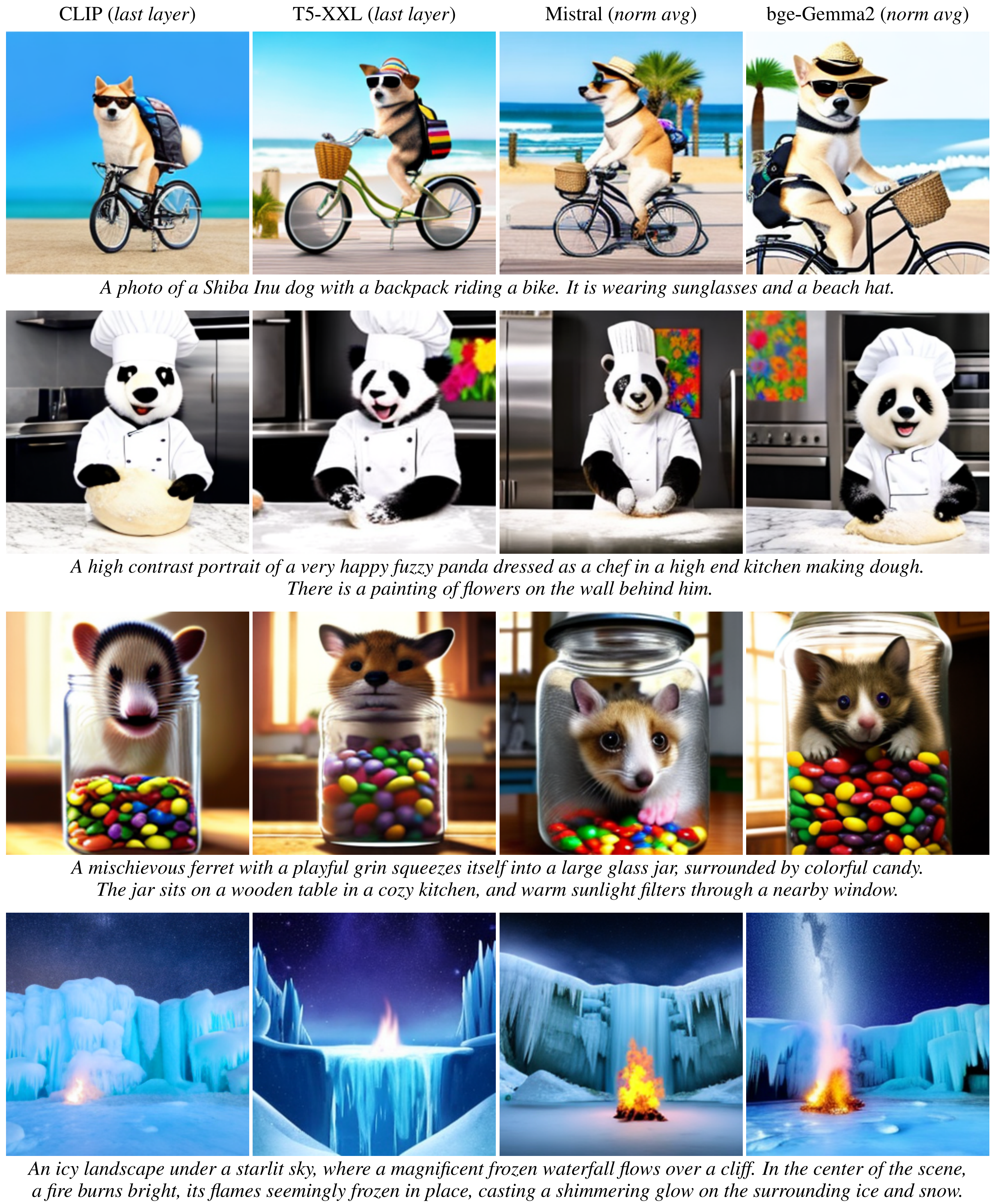

Figure A. Visual comparison of images generated with different text encoders. We use last-layer embeddings (*last layer*) from the text encoders of CLIP-ViT-H/14 (354M) and T5-XXL (4.7B). We also use average layer-normalized embeddings (*norm avg*) from the pre-trained LLM Mistral-7B (7B) and the fine-tuned embedding model bge-Gemma2 (bge-multilingual-gemma2; 9B).

Figure B. Visual comparison of images generated with different text encoders. We use last-layer embeddings (*last layer*) from the text encoders of CLIP-ViT-H/14 (354M) and T5-XXL (4.7B). We also use average layer-normalized embeddings (*norm avg*) from the pre-trained LLM Mistral-7B (7B) and the fine-tuned embedding model bge-Gemma2 (bge-multilingual-gemma2; 9B).

Model	Embeddings	Avg	Attr.	Scene	Spat.	Action	Part	Count.	Comp.	Differ.	Neg.	Uni.
T5-XXL	last layer	0.795	0.795	0.816	0.803	0.803	0.780	0.799	0.800	0.793	0.739	0.804
T5-XXL	norm avg	0.791	0.795	0.810	0.799	0.796	0.771	0.795	0.813	0.783	0.730	0.799
bge-Gemma2	last layer	0.786	0.782	0.808	0.790	0.790	0.774	0.786	0.773	0.781	0.750	0.792
bge-Gemma2	avg	0.809	0.807	0.822	0.815	0.814	0.793	0.811	0.813	0.794	0.755	0.807
bge-Gemma2	norm avg	0.801	0.801	0.821	0.806	0.806	0.790	0.803	0.805	0.789	0.758	0.813
Mistral-7B	last layer	0.782	0.780	0.810	0.790	0.782	0.768	0.786	0.786	0.777	0.738	0.794
Mistral-7B	avg	0.786	0.785	0.809	0.797	0.793	0.774	0.783	0.780	0.761	0.726	0.788
Mistral-7B	norm avg	0.799	0.798	0.820	0.808	0.810	0.789	0.791	0.796	0.790	0.734	0.805

Table E. Original VQAScore for models using different embedding strategies: standard last-layer embeddings (*last layer*), average embeddings across all layers (*avg*), and average embeddings across all normalized layers (*norm avg*). We evaluate the encoder from T5-XXL (4.7B), the pre-trained LLM Mistral-7B (7B), and the fine-tuned embedding model bge-Gemma2 (bge-multilingual-gemma2; 9B).

Model	Avg	Attr.	Scene	Spat.	Action	Part	Count.	Comp.	Differ.	Neg.	Uni.
bge-Gemma2	0.786	0.782	0.808	0.790	0.790	0.774	0.786	0.773	0.781	0.750	0.792
bge-Gemma2$_{\text{pooled}}$	0.802	0.801	0.828	0.807	0.806	0.789	0.812	0.806	0.799	0.763	0.835
sfr-Mistral	0.782	0.780	0.810	0.790	0.782	0.768	0.786	0.786	0.777	0.738	0.794
sfr-Mistral$_{\text{pooled}}$	0.782	0.778	0.810	0.779	0.788	0.774	0.778	0.772	0.772	0.739	0.796

Table F. Original VQAScore for models using embeddings extracted from the last layer compared to models with additional conditioning on global pooled embeddings [6]. We evaluate the fine-tuned embedding models bge (bge-multilingual-gemma2; 9B) and sfr (SFR-Embedding-2_R; 7B).

Model	Size	Avg	Attr.	Scene	Spat.	Action	Part	Count.	Comp.	Differ.	Neg.	Uni.
Qwen2	1.5B	0.758	0.759	0.784	0.761	0.760	0.738	0.756	0.753	0.747	0.704	0.755
Qwen2	7B	0.772	0.772	0.798	0.777	0.782	0.761	0.760	0.759	0.748	0.712	0.785
Gemma2	2B	0.770	0.773	0.797	0.774	0.774	0.759	0.783	0.764	0.756	0.710	0.787
Gemma2	9B	0.782	0.782	0.801	0.790	0.787	0.776	0.781	0.779	0.769	0.708	0.784

Table G. Original VQAScore for models with different LLM sizes, using embeddings extracted from the last layer. We evaluate the pre-trained LLMs: Qwen2 (1.5B), Qwen2 (7B), Gemma2 (2B), Gemma2 (9B).

References

[1] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022. 1

[2] Yunhao Ge, Xiaohui Zeng, Jacob Samuel Huffman, Tsung-Yi Lin, Ming-Yu Liu, and Yin Cui. Visual fact checker: enabling high-fidelity detailed caption generation. In *CVPR*, 2024. 1

[3] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2

[4] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, 2024. 1, 2

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. 1

[6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 6

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[9] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1

[10] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 1

[11] The Mosaic ML Team. Stable diffusion training with mosaicml. *Mosaic Research Blog*, 2023. 1

[12] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. *GitHub repository*, 2022. 1

[13] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023. 1